Python+Selenium

2020年1月6日

20:09

selenium+Python环境配置

1.安装Python开发环境

2.安装selenium pip install selenium

3.安装webdriver(浏览器驱动)

注意:webdriver需要对应的浏览器版本以及selenium版本对应

webdriver 要放到Python的安装目录下才生效.

#导入selenium的webdriver驱动

fromseleniumimportwebdriver

#生命一个浏览器对象

browser=webdriver.Chrome()

#在浏览器中打开该url(一定要加上http或者https不然会报错)

url="https:\\baidu.com"

browser.get(url)

#定位方法

'''

id:find\_element\_by\_id()

name:find\_element\_by\_id\_name

classname:find\_element\_by\_class\_name()

tagname:find\_element\_by\_tag\_name()

linktext:find\_element\_by\_link\_text()

partiallinktext:find\_element\_by\_partial\_link\_text()

xpath:find\_element\_by\_xpath()

cssselector:find\_element\_by\_css\_selector()

'''

'''

browser.find\_element\_by\_id("kw")

browser.find\_element\_by\_name("wd")

browser.find\_element\_by\_class\_name("s\_ipt")

browser.find\_element\_by\_tag\_name("input")

browser.find\_element\_by\_xpath("/html/body/div[1]/div[1]/div/div[1]/div/form/span[1]/input")

browser.find\_element\_by\_link\_text("新闻")

browser.find\_element\_by\_css\_selector("#kw")

#在浏览器开发者模式下选中要定位的位置右键选择Copy选择需要的方式,有xpath和fullXpth和cssselector

'''

#元素操作

'''

clear()：清除文本

send\_keys(value)：模拟按键输入

click()：单击元素

browser.find\_element\_by\_id("kw").clear()

browser.find\_element\_by\_id("kw").send\_keys("selenium")

browser.find\_element\_by\_id("su").click()

submit()提交

search\_text.submit()

search\_text=browser.find\_element\_by\_id('kw')

search\_text.send\_keys('selenium')

search\_text.submit()

'''

#鼠标事件

'''

在WebDriver中,将这些关于鼠标操作的方法封装在ActionChains类中.

fromselenium.webdriverimportActionChains

导入提供鼠标操作的ActionChains类

perform()：执行所有ActionChains中存储的行为

context\_click()：右击

double\_click()：双击

drag\_and\_drop()：拖动

move\_to\_element()：鼠标悬停

#定位到要悬停的元素

above=driver.find\_element\_by\_link\_text("设置")

#对定位到的元素执行鼠标悬停操作

ActionChains(driver).move\_to\_element(above).perform()

ActionChains(driver)调用ActionChains()类，将浏览器驱动driver作为参数传入

move\_to\_element(above)context\_click()方法用于模拟鼠标右键操作在调用时需要指定元素定位

perform()执行所有ActionChains中存储的行为可以理解成是对整个操作的提交动作

'''

#键盘事件

'''

fromselenium.webdriver.common.keysimportKeys

导入提供键盘输入的Keys类

send\_keys(Keys.BACK\_SPACE)删除键（BackSpace）

send\_keys(Keys.SPACE)空格键(Space)

send\_keys(Keys.TAB)制表键(Tab)

send\_keys(Keys.ESCAPE)回退键（Esc）

send\_keys(Keys.ENTER)回车键（Enter）

send\_keys(Keys.CONTROL,'a')全选（Ctrl+A）

send\_keys(Keys.CONTROL,'c')复制（Ctrl+C）

send\_keys(Keys.CONTROL,'x')剪切（Ctrl+X）

send\_keys(Keys.CONTROL,'v')粘贴（Ctrl+V）

send\_keys(Keys.F1)键盘F1...F12

'''

#断言信息

'''

title：用于获得当前页面的标题

current\_url：用户获得当前页面的URL

text：获取搜索条目的文本信息

'''

#设置元素等待

'''

WebDriver提供了两种类型的等待：显式等待和隐式等待

显示等待

显式等待使WebdDriver等待某个条件成立时继续执行，否则在达到最大时长时抛出超时异常(TimeoutException)

element=WebDriverWait(driver,5,0.5).until(

EC.presence\_of\_element\_located((By.ID,"kw"))

)

WebDriverWait类是由WebDirver提供的等待方法在设置时间内，默认每隔一段时间检测一次当前页面元素是否存在，如果超过设置时间检测不到则抛出异常具体格式如下：

WebDriverWait(driver,timeout,poll\_frequency=0.5,ignored\_exceptions=None)

driver：浏览器驱动

timeout：最长超时时间，默认以秒为单位

poll\_frequency：检测的间隔（步长）时间，默认为0.5S

ignored\_exceptions：超时后的异常信息，默认情况下抛NoSuchElementException异常

WebDriverWait()一般由until()或until\_not()方法配合使用，下面是until()和until\_not()方法的说明

until(method,message=‘’)

调用该方法提供的驱动程序作为一个参数，直到返回值为True

until\_not(method,message=‘’)

调用该方法提供的驱动程序作为一个参数，直到返回值为False

隐式等待:

WebDriver提供了implicitly\_wait()方法来实现隐式等待，默认设置为0。它的用法相对来说要简单得多

driver.implicitly\_wait(10)

try:

print(ctime())

driver.find\_element\_by\_id("kw22").send\_keys('selenium')

exceptNoSuchElementExceptionase:

print(e)

finally:

print(ctime())

implicitly\_wait()默认参数的单位为秒，本例中设置等待时长为10秒首先这10秒并非一个固定的等待时间，它并不影响脚本的执行速度其次，它并不针对页面上的某一元素进行等待当脚本执行到某个元素定位时，如果元素可以定位，则继续执行；如果元素定位不到，则它将以轮询的方式不断地判断元素是否被定位到假设在第6秒定位到了元素则继续执行，若直到超出设置时长（10秒）还没有定位到元素，则抛出异常

'''

#定位一组元素

'''

find\_elements\_by\_id()

find\_elements\_by\_name()

find\_elements\_by\_class\_name()

find\_elements\_by\_tag\_name()

find\_elements\_by\_link\_text()

find\_elements\_by\_partial\_link\_text()

find\_elements\_by\_xpath()

find\_elements\_by\_css\_selector()

定位一组元素的方法与定位单个元素的方法类似，唯一的区别是在单词element后面多了一个s表示复数

'''

#多表单切换

'''

在Web应用中经常会遇到frame/iframe表单嵌套页面的应用，WebDriver只能在一个页面上对元素识别与定位，对于frame/iframe表单内嵌页面上的元素无法直接定位这时就需要通过switch\_to.frame()方法将当前定位的主体切换为frame/iframe表单的内嵌页面中

进入到界面后并不是账号密码登录界面,需要点击切换一下,

driver.find\_element\_by\_id("lbNormal").click().

不能通过id或者name定位,id是随机变化的,每次刷新页面都会获取到一个新的id,所以这里用fullxpath来定位.

xf=driver.find\_element\_by\_xpath("/html/body/div[2]/div[3]/div/div[3]/div[4]/div[1]/div/iframe")

driver.switch\_to.frame(xf)

driver.find\_element\_by\_name("email").clear()

driver.find\_element\_by\_name("email").send\_keys("username")

driver.find\_element\_by\_name("password").clear()

driver.find\_element\_by\_name("password").send\_keys("password")

driver.find\_element\_by\_id("dologin").click()

driver.switch\_to.default\_content()

'''

#多窗口切换

'''

在页面操作过程中有时候点击某个链接会弹出新的窗口，这时就需要主机切换到新打开的窗口上进行操作WebDriver提供了switch\_to.window()方法，可以实现在不同的窗口之间切换

#获得百度搜索窗口句柄

sreach\_windows=driver.current\_window\_handle

driver.find\_element\_by\_link\_text('登录').click()

driver.find\_element\_by\_link\_text("立即注册").click()

#获得当前所有打开的窗口的句柄

all\_handles=driver.window\_handles

#进入注册窗口

forhandleinall\_handles:

ifhandle!=sreach\_windows:

driver.switch\_to.window(handle)

print('nowregisterwindow!')

driver.find\_element\_by\_name("userName").send\_keys('username')

driver.find\_element\_by\_name('password').send\_keys('password')

time.sleep(2)

current\_window\_handle：获得当前窗口句柄

window\_handles：返回所有窗口的句柄到当前会话

switch\_to.window()：用于切换到相应的窗口，与上一节的switch\_to.frame()类似，前者用于不同窗口的切换，后者用于不同表单之间的切换

'''

#警告框处理

'''

在WebDriver中处理JavaScript所生成的alert、confirm以及prompt十分简单，具体做法是使用switch\_to.alert方法定位到alert/confirm/prompt，然后使用text/accept/dismiss/send\_keys等方法进行操作

text：返回alert/confirm/prompt中的文字信息

accept()：接受现有警告框

dismiss()：解散现有警告框

send\_keys(keysToSend)：发送文本至警告框keysToSend：将文本发送至警告框

switch\_to\_alert()方法接受这个弹窗

driver.switch\_to.alert.accept()

通过switch\_to\_alert()方法获取当前页面上的警告框，并使用accept()方法接受警告框

'''

#下拉框选择

'''

WebDriver提供了Select类来处理下拉框如百度搜索设置的下拉框

fromselenium.webdriver.support.selectimportSelect

sel=driver.find\_element\_by\_xpath("//select[@id='nr']")

Select(sel).select\_by\_value('50')#显示50条

select类用于定位select标签

select\_by\_value()方法用于定位下接选项中的value值

'''

#关闭浏览器

browser.close()